

AMENDMENTS TO THE SPECIFICATION

Please insert the following paragraph before the paragraph beginning on page 3, line 1:

--

SUMMARY OF THE INVENTION

A method for transferring data is disclosed. In some embodiments, a synchronization operation uses a first processing thread to copy a first file from a first data storage to a second data storage, and a second processing thread to copy a second file from the first data storage to the second data storage. The first file is copied substantially concurrently with the copying of the second file, and the first operation results in a first copied file and a second copied file in the second data storage. A real-time replication operation is performed wherein the operation updates the first copied file and the second copied file in an order determined at least in part by an order in which changes were made to the first file and the second file, respectively, as stored in the first data storage.

--

Please replace the paragraph beginning on page 4, line 20, with the following rewritten paragraph:

--

Figure 1 is a block diagram of a data replication system according to an embodiment of the present invention. In this example, a source machine 100 is coupled with a target machine 104. The source machine 100 and the target 104 are ~~showed~~ shown to be coupled by a plurality of connections 100A-100C, 100A'-100C' through a network 102, such as a LAN or WAN.

--

Please replace the paragraph beginning on page 6, line 7, with the following rewritten paragraph:

--

During a sync operation if no real-time replication is also occurring, the kernel cache will generally by file be filled with SYNC_COPY commands. In this case, all site threads (main site thread as well as sync threads) will run down the queue, skipping all SYNC_COPY commands that have been or are being processed by other threads until it finds one that has not yet been processed. The thread will then perform whatever operation is requested for that command (full, incremental, compare attributes, etc.). If a sync thread encounters a non-sync command that has not yet been processed, it will go into a wait state until the [[mail]] main site thread has caught up with it and processed the non-sync command.

--
Please replace the paragraph beginning on page 8, line 4, with the following rewritten paragraph:

--
In this example, the thread retrieves the next command from the queue (200). It is determined whether all sync threads have sent configuration to the target machine (202). The configuration is sent through the thread to the target machine when the sync thread tries to connect to the target machine in order to inform the target machine what it is sending. If all sync threads have not yet [[send]] sent the configurations to the target, then the thread goes into a wait state and does not process the queue entry and does not move to the next entry (212). Once all of the sync threads have sent their configurations to the target (202), then it is determined whether this particular queue entry is a sync command (204). If it is a sync command, then it is determined whether there is another thread processing this entry (206). If there is no other thread processing this entry, then the entry is marked as now being processed (208). The thread then continues to process the entry (210). For example, the file associated with the entry is transferred from the source machine to the target machine.